

Я.Ю. КОРОЛЕВА, асс. ХНУРЭ, **М.Г. РЫЖИКОВА**, асс. ХНУРЭ

СИНТЕЗ ТЕСТОВ ДЛЯ СЕТЕЙ КЛЕТОЧНЫХ АВТОМАТОВ НА ОСНОВЕ ХАРАКТЕРИСТИЧЕСКИХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Пропонується метод та алгоритм синтезу тестів для мереж клітинних автоматів, побудованих на використанні характеристичних та синхронізуючих послідовностей автоматних моделей клітин мережі. Отримана верхня межа довжини перевіряючих тестів, які виявляють множину несправностей в класу.

The method and algorithm deriving an test sequences for the cellular automata arrays are proposed. The method is based on using synchronizing and characteristic sequences deriving from state table of cellular automata. The greatest lower bound of the test sequences that detect all functional fault of cellular automata arrays is found.

Введение. В последнее время растет число исследований, связанных с однородными сетями, состоящими из однотипных и одинаково соединенных модулей – ячеек. Таким сетям под различными названиями – итеративные сети, однородные среды, систолические сети, сети клеточных автоматов (СКА) и т.д. посвящено большое число исследований [1-7]. Широкое распространение программируемых логических интегральных схем (ПЛИС) типа *FPGA* и *CPLD*, обладающих регулярностью структурной организации, определяет интерес исследователей к реализации вычислительных устройств и систем в виде однородных сетей клеточных автоматов, в которых каждая ячейка или клеточный автомат (КА) имеет ограниченное множество состояний (от 2 до 32). Сети клеточных автоматов являются дискретными динамическими системами, поведение которых полностью определяется правилами переходов состояний КА в зависимости от состояний его близких соседей.

Различают одномерные и двумерные сети клеточных автоматов. Наибольшее практическое применение нашли структуры клеточных автоматов, у которых клеточные автоматы имеют два состояния [2, 8, 11]. В одномерных СКА клеточный автомат имеет достаточно простую структуру и состоит из триггера и комбинационной схемы, реализующей функцию возбуждения, равную $z_i(t+1) = f_i(z_{(i-1)}(t), z_i(t), z_{(i+1)}(t))$, где $z_{(i-1)}(t), z_{(i+1)}(t)$ – текущее состояние соседних КА, расположенных слева и справа от ячейки z_i , соответственно. Функцию f называются правилом настройки КА. Так как f зависит от трех переменных, то существует $2^3 = 256$ различных правил настройки, многие из которых являются тривиальными логическими функциями. Номер правила – это десятичное значение восьмиразрядного двоичного числа, представляющего функцию f .

Сеть клеточных автоматов, все клетки которой настроены по одному и тому же правилу, называют однородной СКА, в противном случае она называется гибридной СКА. Однородная или гибридная СКА являются аддитивной СКА, если правила настройки клеток является классом линейных булевых функций [8].

Одномерные гибридные СКА широко используются в качестве генераторов псевдослучайных последовательностей и сигнатурных анализаторов [12,13,14,17,22]. Достоинствами таких устройств является простота и регулярность структуры, а увеличение разрядности генераторов связано лишь с добавлением однотипных ячеек. В таких СКА отсутствуют глобальные обратные связи, которые характерны для генераторов на сдвиговых регистрах с нелинейными обратными связями.

Использование двумерных СКА для решения многих задач распознавания образов и связанных с ними проблемами, моделирование гидродинамических процессов и других физических явлений, в основе которых лежит статистическая механика, демонстрирует потенциальные возможности СКА с учетом простоты их схемной реализации на СБИС [6,7,9]. В качестве примера этой демонстрации можно привести СКА, структуру которого и систему правил определил в 1970 г. известный математик Дж. Конвей для игры в «Жизнь» [1]. В других физических приложениях СКА использовались для моделирования роста кристаллов, где было показано, что анализ поведения этой сети позволил уточнить физическую сущность этих сложных процессов [7,18,21]. В большинстве опубликованных работ для анализа поведения одномерных и двумерных СКА использовался аппарат алгебры многочленов.

Эффективность использования СКА для моделирования сложных динамических систем достигается параллельным взаимодействием большого числа переменных (одна на клетку) при ограничении их локальными связями и единообразными правилами эволюции. Для некоторых научных проблем в эксперименте необходимо вычислять миллиард событий (событием является обновление одной клетки), для более сложных приложений это число может увеличиться в тысячу или миллион раз (т.е. 10^{12} - 10^{15} событий) [1]. Для того, чтобы вычислить 10^{13} событий на современном быстродействующем компьютере потребуется 5-8 месяцев. Для сравнения двумерная сеть КА, содержащая 10^5 клеточных автоматов (это несколько грубее, чем разрешающая способность телевизионного кадра) выполнит моделирование 10^{13} событий за 0,5 сек [1, 18].

Однако отсутствие в СКА доступа к внутренним ячейкам сети создает значительные проблемы при верификации правильности настроек КА, проверке исправности всей сети и локализации места неисправной ячейки СКА. В [16] представлен метод синтеза проверяющих тестов для класса одномерных сетей клеточных автоматов СКА, в которых наблюдаемыми выходами являются только боковые выходы граничных ячеек сети. Метод основан на использовании автоматных моделей ячеек СКА и функциональных моделях

неисправностей. Показано, что сеть является тестируемой относительно класса неисправностей F_1 , который включает полное множество константных неисправностей ячейки, перепутываний, инверсий, подкласс перемычек и коротких замыканий, не увеличивающих числа состояний автоматной модели ячейки сети. Кроме того, неисправности класса F_1 являются устойчивыми на время прохождения проверяющего теста и допускается в момент проверки сети неисправной лишь одна произвольная ячейка сети. Предложено и обосновано применение циклических отличительных последовательностей для построения проверяющего эксперимента СКА.

Если автоматная модель ячейки СКА не содержит отличительных, символов, то проверка правильности всех переходов в каждой ячейке, усложняется трудоемкостью транспортировки проверяемого состояния произвольной ячейки СКА на единственный наблюдаемый боковой выход сети. Однако, не все циклические переводящие последовательности можно использовать в качестве транспортирующих. Кроме того, эффективность синтезированных тестов, обнаруживающих класс одиночных константных неисправностей, ниже эффективности тестов, обнаруживающих класс произвольных неисправностей ячейки сети (класс F_1).

В статье представлен метод построения полного проверяющего, эксперимента в одномерной СКА без наблюдаемых выходов в каждой ячейке сети, обнаруживающего множество неисправностей класса F_1 , который основан на функциональном подходе и использовании характеристических и синхронизирующих последовательностей автоматной модели ячейки сети.

Характеристические и синхронизирующие последовательности автоматов. Известно, что большинство реальных схем СКА, используемых на практике, состоит из ячеек, поведение которых на функциональном уровне описывается моделью минимального автомата Мура. В общем случае, любой минимальный автомат Мура имеет множество характеристических символов, которые также, как и характеристические последовательности автоматных моделей ДУ с элементами памяти, можно использовать при построении ДЭ в одномерных СКА, не имеющих отличительных символов [16]. При этом подходе правильность каждого перехода ячейки сети проверяется не одним тестовым набором, а несколькими наборами, число которых определяется числом характеристических символов автоматной модели ячейки сети. В [20] приведены методы синтеза синхронизирующих последовательностей (СП) по таблице переходов-выходов автомата и по функциональной схеме ДУ.

Определение 1. Входная последовательность X_s автомата, которая устанавливает его в определенное конечное состояние независимо от состояния выхода и начального состояния, называется синхронизирующей последовательностью.

Если автомат $A(X, Z, Y, d, I)$ задан таблицей переходов-выходов, то из определения 1 следует, что автомат имеет синхронизирующую последовательность тогда и только тогда, когда существует входная последовательность X_s такая, что $d(z_i, X_s) = z_0$, $\forall z_i \in Z$, $z_0 \in Z$. Множество переходов $d(z_i, X_s) = z_0$, $\forall z_i \in Z$, автомата определяет отображение множества его состояний Z в некоторое определенное состояние z_0 при подаче на автомат входной последовательности X_s , то есть $z_i \xrightarrow{X_s} z_0$, $i = \overline{1, n}$.

Синхронизирующая последовательность для заданного автомата может быть найдена из синхронизирующего дерева, которое является деревом-преемником, построенным по определенным правилам, представленным в [20]. Необходимые условия, которым должен удовлетворять автомат, обладающий синхронизирующей последовательностью, определяются следующей теоремой.

Теорема 1. Автомат имеет синхронизирующую последовательность, если существует, по крайней мере, один входной символ $x_k \in X$ такой, что

$$d(z_i, x_k) = d(z_j, x_k), \text{ где } i \neq j, 1 \leq i, j \leq n, n \geq 2.$$

Доказательство. Если условие теоремы не выполняется, то множество состояний Z автомата будет отображаться в то же множество для всех входных символов и на первом шаге построение синхронизирующего дерева завершится в соответствии с правилом 1 построения синхронизирующего дерева [20].

Метод синтеза проверяющих тестов на основе использования характеристических и синхронизирующих последовательностей рассмотрим на примере одномерной сети, ячейка которой представлена таблицей переходов-выходов в табл. 1 (пример 4.4 в [19]).

Таблица 1 – ТПВ ячейки

$z(t)$	$z(t+1)$	
	x_1	x_2
z_1	z_1	z_2
z_2	z_1	z_4
z_3	z_2	z_2
z_4	z_2	z_4

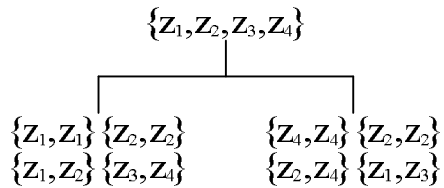


Рис. 1. Характеристическое дерево автомата

Как видно из характеристического дерева на рис. 1 автоматная модель ячейки не имеет отличительных символов. Так как $p_{x_1} \times p_{x_2} = p(0)$, где $p_{x_1} = \{\overline{z_1 z_2}; \overline{z_3 z_4}\}$, а $p_{x_2} = \{\overline{z_1 z_3}; \overline{z_2 z_4}\}$, то имеется два характеристических

символа $X_c = \{x_1, x_2\}$, позволяющих различать все множество состояний автомата по состоянию наблюдаемого выхода.

Следовательно, полный проверяющий тест для СКА из ячеек такого типа не может быть построен методом, рассмотренном в [16]. В [19] при построении проверяющих тестов для такой сети использовалась модель одиной константной неисправности. Если ограничиться числом ячеек сети $p = 4$, то в [19] было найдено 12 тестов для сети из ячеек табл. 1, которые позволяют обнаружить множество константных неисправностей на соединениях каждой ячейки сети.

В отличие от рассмотренного выше подхода поставим задачу нахождения проверяющего теста, обнаруживающего в заданной СКА множество функциональных неисправностей ячейки сети при условии, что в момент проверки неисправной может быть только одна ячейка (класс неисправностей F_1). Для анализа свойств ячейки построим синхронизирующее дерево – префиксов ее автоматной модели (рис. 2), из которого следует, что ячейка сети имеет две синхронизирующие последовательности $X_s = \{(x_1, x_1), (x_2, x_2)\}$.

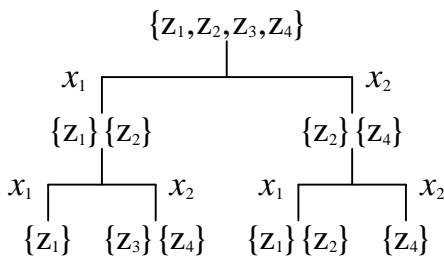


Рис. 2. Синхронизирующее дерево автомата

Как видно из синхронизирующего дерева рис. 2 путь от корня дерева к висячей вершине, отмеченной одним состоянием, сопровождается постоянным уменьшением, мощности начального S -множества, что эквивалентно уменьшению различимости множества начальных состояний. С другой стороны, путь, не приводящий к висячей вершине с одним состоянием, например $(x_1 x_2 x_1)$ или $(x_2 x_1 x_2)$ на рис. 2, начинающийся с характеристического символа, приводит к вершине, отмеченной S -множеством, совпадающим с S -множеством меньшего ранга синхронизирующего дерева, что определяет отношение эквивалентности состояний в этих вершинах.

Определение 2. Путь в синхронизирующем дереве автоматной модели ячейки сети, начинающийся с характеристического символа и не содержащий синхронизирующей последовательности, будем называть характеристическим путем.

Утверждение 1. Пусть состояние z_j ячейки $C(k)$ СКА транспортируется на боковой наблюдаемый выход сети приложением входного набора X_T к входам ячеек $C(k+1), C(k+2), \dots, C(p)$. Входной набор X_T является проверяю-

щим тестом, отличающим на наблюдаемом выходе состояние z_j от некоторого подмножества состояний $Z' \subset Z$, если последовательность символов X_T является характеристическим путем в синхронизирующем дереве-преемников автоматной диаграммы ячейки.

Возвращаясь к рассматриваемому примеру сети, на основании утверждения 1 последовательность X_T , которая транспортирует неисправность на правый крайний выход сети, не должна содержать повторяющиеся символы x_1 и x_2 , так как x_1x_1 и x_2x_2 образуют синхронизирующую последовательность. Как видно из синхронизирующего дерева-преемников (рис. 2) для данной ячейки в качестве транспортирующих можно использовать последовательности вида: $(x_1x_2x_1\dots)$ и $(x_2x_1x_2\dots)$, не содержащие X_s . Действительно, так как автомат не имеет отличительных символов, то приложение любого входного символа x_1 или x_2 не позволяет различить лишь пары совместимых состояний на выходе последующей ячейки, а последующее приложение другого входного символа, отличающегося от предыдущего, не увеличивает степень неразличимости состояний.

Пусть сеть состоит из четырех ячеек. Построим тест, проверяющий переход $z_1 \xrightarrow{x_1} z_1$ в первой ячейке сети, в виде

$$z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \quad (1)$$

В соответствии с утверждением 1 состояние z_1 на входе $C(2)$ транспортируется на правый крайний выход последовательностью $(x_1x_2x_1)$, которая не содержит синхронизирующей. Однако тест (1) не является полным тестом, различающим состояние z_1 на входе $C(2)$ от множества других. Так, ошибка

типа $z_1 \xrightarrow{x_1} z_2$ не различается этим тестом $z_1 \xrightarrow{x_1} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1$ ввиду того, что пара состояний (z_1, z_2) является x_1 -совместимой. Для обнаружения этой ошибки используется тестовый набор вида

$$z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \quad (2)$$

Так как x_2 различает пару состояний (z_1, z_2) , то неисправность $z_1 \xrightarrow{x_1} z_2$

будет обнаружена на правом выходе сети: $z_1 \xrightarrow{x_1} z_2 \xrightarrow{x_1} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4$.

Таким образом, два тестовых набора (1) и (2) позволяют обнаружить полное множество неисправностей типа $(z_1 - Z/z_1)$ на выходе первой ячейки сети. Следует отметить, что для построения этих тестов использовалось пол-

ное множество характеристических символов $X_s = \{x_1 x_2\}$ автоматной модели ячейки, с которых начинается формирование транспортирующих последовательностей X_T , удовлетворяющих условию Утверждения 1.

Проведенный выше анализ позволяет определить необходимые и достаточные условия проверки исправности любого перехода в автоматной модели ячейки сети, не имеющей отличительных символов.

Теорема 2. Пусть ячейка однородной сети имеет множество характеристических символов $X_s = \{x_1, x_2, \dots, x_r\}$. Для проверки правильности перехода

$z_i \xrightarrow{x_k} z_j$ в ячейке $C(k)$ сети построим множество входных наборов

$X_k^i = \{x_{k+1}^i, x_{k+1}, \dots, x_p\}$, $i = \overline{1, r}$, которые прикладываются к верхним входам

ячеек $C(k+1)$, $C(k+2)$, ..., $C(p)$. Если в последовательности X_k^i каждый пер-

вый символ $\{x_{k+1}^i \in X_c, x_{k+1}^i \neq x_{k+1}^{i+1}, i = \overline{1, r}$ и каждая последовательность X_k^i

является характеристическим путем, то множество наборов X_k^i является

полным проверяющим тестом перехода $z_i \xrightarrow{x_k} z_j$ ячейки $C(k)$.

Следствие 1. Пусть ячейка СКА без наблюдаемых выходов представляется моделью минимального автомата Мура, имеющего n состояний и m входных символов. Если число ячеек сети равно p , то верхняя граница числа проверяющих тестов не превышает $[(p-1) \cdot (n-1) + 1] \cdot m \times n$ для класса F_1 обнаруживаемых неисправностей.

Доказательство. Из условия минимальности автоматной модели ячейки сети существует не более $(n-1)$ характеристических символов, которые в одной ячейке сети позволяют получить $(n-1)$ входных наборов, проверяющих правильность одного перехода в автоматной диаграмме. Так как число переходов равно $m \times n$, а число ячеек p , то потребуется не более $(p-1)(n-1) m \times n$ входных наборов для проверки правильности всех $m \times n$ переходов в первых $(p-1)$ ячейках сети.

Так как выход последней ячейки $C(p)$ -наблюдаемый, то достаточно проверить $m \times n$ переходов в этой ячейке по состоянию наблюдаемого выхода. В результате, полное число проверяющих наборов не превышает $(p-1)(n-1)m \cdot n + mn = [(p-1)(n-1) + 1] \cdot mn$.

В реальных устройствах число проверяющих тестов, обнаруживающих класс неисправностей F_1 , значительно меньше полученной выше верхней границы длины проверяющего эксперимента. Это объясняется тем, что тест, построенный для проверки определенного перехода в некоторой ячейке, одновременно проверяет некоторые переходы в других ячейках.

Возвращаясь к рассматриваемому примеру сети из ячеек табл.1, можно заметить, что тест (1), проверяющий переход $z_1 \xrightarrow{x_1} z_1$ в первой ячейке сети $C(1)$, одновременно проверяет правильность этого перехода в ячейке $C(2)$ Различие заключается лишь в том, что в $C(1)$ проверка осуществляется характеристическим символом x_1 , а в $C(2)$ -символом x_2 .

Множество переходов автоматной диаграммы ячейки (таблица 1) обозначим следующим образом:

$$\begin{array}{llll} 1: & z_1 \xrightarrow{x_1} z_1, & 3: & z_2 \xrightarrow{x_1} z_1, & 5: & z_3 \xrightarrow{x_1} z_2, & 7: & z_4 \xrightarrow{x_1} z_2, \\ 2: & z_1 \xrightarrow{x_2} z_2, & 4: & z_2 \xrightarrow{x_2} z_4, & 6: & z_4 \xrightarrow{x_2} z_2, & 8: & z_4 \xrightarrow{x_2} z_4. \end{array}$$

Тогда тест (1) проверяет множество переходов $\{1/x_1 - C(1); 1/x_2 - C(2); 2/x_1 - C(3); 3/x_1 - C(4)\}$, где $\{1/x_1 - C(1)\}$ —обозначает, что переход 1 проверяется по характеристическому символу x_1 в ячейке $C(1)$; $\{3/_- - C(4)\}$ —переход 3 проверяется в ячейке $C(4)$.

Так как автоматная модель ячейки имеет два характеристических символа x_1 и x_2 , то в соответствии с теоремой 2 для каждого перехода необходимо иметь два тестовых набора, проверяющих его в каждой ячейке по двум характеристическим символам. Причем каждый такой набор является характеристическим путем в синхронизирующем дереве-приемнике автоматной модели ячейки.

Так как часть характеристического пути является также характеристическим путем, то проверяющий тест, построенный для предшествующей ячейки сети является также проверяющим тестом для последующих ячеек сети, если каждый символ проверяющего набора является характеристическим. Поэтому целесообразно начинать построение тестовых наборов для ячеек в последовательности $C(1), C(2), \dots, C(p)$.

Также как в экспериментах с легко тестируемыми СКА, при установке в проверяемой ячейке требуемого состояния можно использовать любую переводящую последовательность, к которой не предъявляется никаких дополнительных требований ввиду принятых допущений на класс неисправностей F_1 .

Алгоритм синтеза проверяющих тестов.

Рассмотренный выше функциональный подход, основанный на использовании автоматных моделей ячеек СКА и теории экспериментов над автоматами, позволяет представить процедуру синтеза полных проверяющих последовательностей, обнаруживающих неисправности класса F_1 в СКА с боковыми наблюдаемыми выходами, не имеющих отличительных символов, в следующем виде.

Алгоритм 1.

1. Построить модель автомата Мура ячейки сети по ее схеме или правилу настройки.

2. Построить характеристическое дерево автомата и найти множество характеристических символов $X_c = \{x_1, x_2, \dots, x_r\}$.

3. Построить синхронизирующее дерево-преемников автомата и найти множество характеристических путей X_T^i для каждого характеристического символа x_i , $i = \overline{1, r}$.

4. Установить номер проверяемой ячейки $C(i)$ $I = 1$ проверяемого перехода $j = 1$.

5. Синтезировать множество из r тестовых наборов, проверяющих правильность j -го перехода в i -ой ячейке сети путем использования множества X_c , определенного на шаге 2, и множества X_T^i , определенного на шаге 3, с учетом множества тестов, полученных для ячеек с номерами $(i-1)(i-2)\dots 1$.

6. Определить множество переходов, проверяемых тестами, полученными на шаге 5, в ячейках с номерами $(i+1)(i+2)\dots p$.

7. Установить $j = j+1$. Если $j \leq mn$, то перейти к шагу 5. В противном случае установить $j+1$ и перейти к шагу 8.

8. Конец алгоритма.

Синтез проверяющих тестов в соответствии с алгоритмом 1 для СКА с наблюдаемыми боковыми выходами иллюстрируется на примере сети из четырех ячеек (табл. 1). Множество проверяющих тестов, обнаруживающих в каждой ячейке неисправности переходов ее автоматной диаграммы типа $z_i - Z / z_i$, приведено на рис.3. Правильность каждого перехода проверяется двумя тестовыми наборами, транспортирующими проверяемое состояние на наблюдаемый выход последовательностями, которые в соответствии с теоремой 2 начинаются с характеристического символа и каждая последовательность является характеристическим путем в синхронизирующем дереве-преемников автоматной модели ячейки сети.

Справа от каждого тестового набора приведено множество переходов, которые одновременно проверяются данным тестом. Это множество определяется при выполнении шага 6 алгоритма. Как видно из рис. 3, длина проверяющего теста равна 28. Заметим, что в [19] множество тестов для аналогичной сети, обнаруживающих класс одиночных константных неисправностей, включает 12 тестов. Увеличение длины, проверяющей последовательности при использовании алгоритма 1 синтеза проверяющих тестов является платой за значительное расширение класса обнаруживаемых неисправностей, принятого при функциональном подходе к построению диагностического эксперимента.

Тесты
C(1) C(2) C(3) C(4)

$$t_1 : z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1$$

$$t_2 : z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2$$

$$t_3 : z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2,$$

$$t_4 : z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1,$$

$$t_5 : z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1$$

$$t_6 : z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4,$$

$$t_7 : z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2,$$

$$t_8 : z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_2} z_4,$$

$$t_9 : z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2,$$

$$t_{10} : z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1,$$

$$t_{11} : z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_2,$$

$$t_{12} : z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_1} z_1$$

$$t_{13} : z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_2,$$

$$t_{14} : z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4,$$

$$t_{15} : z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2,$$

$$t_{16} : z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4,$$

Проверяемые переходы

$$\{1/x_1 - C(1); 1/x_2 - C(2);$$

$$2/x_1 - C(3); 3/-C(4)\}$$

$$\{1/x_2 - C(1); 2/x_1 - C(2);$$

$$3/x_2 - C(3); 2/-C(4)\}$$

$$\{1/x_1 - C(2); 1/x_2 - C(3); 2/-C(4)\}$$

$$\{1/x_1 - C(3); 1-C(4)\}$$

$$\{2/x_1 - C(1); 3/x_2 - C(2);$$

$$2/x_1 - C(3); 3/-C(4)\}$$

$$\{2/x_2 - C(1); 4/x_1 - C(2);$$

$$7/x_2 - C(3); 4/-C(4)\}$$

$$\{2/x_2 - C(2); 4/x_1 - C(3); 7/-C(4)\}$$

$$\{2/x_2 - C(3); 4/-C(4)\}$$

$$\{3/x_2 - C(1); 2/x_1 - C(2);$$

$$3/x_2 - C(3); 2/-C(4)\}$$

$$\{3/x_1 - C(1); 1/x_2 - C(2);$$

$$2/x_1 - C(3); 3/-C(4)\}$$

$$\{3/x_1 - C(2); 1/x_2 - C(3); 2/-C(4)\}$$

$$\{3/x_1 - C(3); 1/-C(4)\}$$

$$\{4/x_1 - C(1); 7/x_2 - C(2);$$

$$4/x_1 - C(3); 7/-C(4)\}$$

$$\{4/x_2 - C(1); 8/x_1 - C(2);$$

$$7/x_2 - C(3); 4/-C(4)\}$$

$$\{4/x_2 - C(2); 8/x_1 - C(3); 7/-C(4)\}$$

$$\{4/x_2 - C(3); 8/-C(4)\}$$

$t_{17} : z_3 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4,$	$\{5/x_2 - C(1); 8/x_1 - C(2);$ $7/x_2 - C(3); 8/-C(4)\}$
$t_{18} : z_3 \xrightarrow{x_1} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1,$	$\{5/x_1 - C(1); 3/x_2 - C(2);$ $2/x_1 - C(3); 4/-C(4)\}$
$t_{19} : z_3 \xrightarrow{x_2} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4,$	$\{6/x_2 - C(1); 4/x_1 - C(2);$ $7/x_2 - C(3); 4/-C(4)\}$
$t_{20} : z_3 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1,$	$\{6/x_1 - C(1); 3/x_2 - C(2);$ $2/x_1 - C(3); 3/-C(4)\}$
$t_{21} : z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4,$	$\{7/x_2 - C(1); 4/x_1 - C(2);$ $7/x_2 - C(3); 4/-C(4)\}$
$t_{22} : z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2 \xrightarrow{x_1} z_1,$	$\{7/x_1 - C(1); 3/x_2 - C(2);$ $2/x_1 - C(3); 3/-C(4)\}$
$t_{23} : z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_1} z_1 \xrightarrow{x_2} z_2,$	$\{7/x_1 - C(2); 3/x_2 - C(2);$ $2/x_1 - C(3); 3/-C(4)\}$
$t_{24} : z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_1} z_1,$	$\{7/x_1 - C(3); 3/-C(4)\}$
$t_{25} : z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2,$	$\{8/x_1 - C(1); 7/x_2 - C(2);$ $4/x_1 - C(3); 7/-C(4)\}$
$t_{26} : z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2 \xrightarrow{x_2} z_4,$	$\{8/x_2 - C(1); 8/x_1 - C(2);$ $7/x_2 - C(3); 4/-C(4)\}$
$t_{27} : z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_1} z_2,$	$\{8/x_2 - C(2); 8/x_1 - C(3); 7/-C(4)\}$
$t_{28} : z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4 \xrightarrow{x_2} z_4,$	$\{8/x_2 - C(3); 8/-C(4)\}$

Рис. 3. Множество проверяющих тестов для сети из ячеек табл. 1

Выводы. Предложен метод и разработан алгоритм синтеза проверяющих тестов для одномерных СКА с ограниченной наблюдаемостью выходов ячеек сети. Метод основан на использовании автоматных моделей ячеек СКА и теории экспериментов над автоматами. Рассмотрен класс сетей, ячейки которых не имеют отличительных символов. Для синтеза проверяющих тестов предложено использовать характеристические и синхронизирующие последовательности. Введено понятие характеристического пути, как пути в синхронизирующем дереве-преемников, начинающегося с характеристического символа и не содержащего синхронизирующей последовательности. Показано, что входной набор, который представляет собой характеристический путь

в синхронизирующем дереве-преемников автоматной диаграммы ячейки ОС, является проверяющим тестом ОС. Для класса СКА, не имеющих отличительных символов, предложен и исследован метод построения полного проверяющего эксперимента, позволяющий обнаружить неисправности класса F_1 . Показано, что увеличение длины проверяющей последовательности в предложенном методе по сравнению с существующими, является платой за значительное расширение класса обнаруживаемых неисправностей, принято при функциональном подходе к построению диагностического эксперимента над автоматами.

Список литературы: 1. *Von Neumann J.* Theory of self Reproducing Automata. Ed.A.W. Burks. – Urbana: University of Illinois Press.–1966.–485 p. 2. *Тофолу Т., Марголуз Н.* Машины клеточных автоматов. – М.: Из-во «Мир», – 1991. – 280с. 3. *Евреинов Э.В., Грангивили И.В.* Цифровые автоматы с настраиваемой структурой. – М.: Энергия, –1974. –240с. 4. *Варшавский В.И., Мараховский В.Б.* Однородные структуры. Анализ. Синтез. Поведение. –М.: Энергия, –1973. –152с. 5. *Mukhopadhyay A.* Unate cellular logic // IEEE Trans. On Computers.–1969. –Vol. C-18, №2.- P.114-121. 6. *Wolfram S.* Computation theory of cellular automata // Communications in Mathematical Physics.–1984.–Vol. 93.–P. 15–57. 7. *Wolfram S.* Statistical mechanics of cellular automata // Reviews of Modern Physics.–1983.–Vol. 55, № 3.–P. 601–644. 8. *Das A.K. et al.* Efficient characterization of cellular automata // IEE Proc. Part (E). -1990.- Vol. 137, №1.-p. 81-87. 9. *Martin O., Odlyzko A.M., and Wolfram S.* Algebraic properties of cellular automata. // Communications in Mathematical Physics.–1984.–Vol. 93.–p. 219. 10. *Gloster C.R., Borglez F.* Boundary scan with cellular–based built–in self– test // Proc. Intern. Test Conference–1988.–p. 138–145. 11. *Gortensius P.D., Card H.C., McLeod R.D., Pries W.* Importance sampling for using computers using one–dimensional cellular automata. // IEEE Trans. Comput.–1989. -Vol.C-38,№6.-p. 769-774. 12. *Gortensius P.D., Card H.C., McLeod R.D.* Parallel random number generation for VLSI systems using cellular automata. // IEEE Trans. Comput.–1989.–Vol. C-38, № 10. -p. 1466-1473. 13. *Neebel D J., Kime C.R.* Cellular automata for weighted random pattern generation // IEEE Trans. On Computers.–1997.–Vol. C-46,№ 11.–P. 1219–1228. 14. *Hortensius P.D., McLeod R.D., Card H.C.* Cellular automata-based signature analysis for built-in self-test // IEEE Trans. On Computers.–1990.–Vol. C-39, № 10. -P. 1273-1283. 15. *Дербунович Л.В., Бережная М.А., Королева Я.Ю., Рыжикова М.Г.* Тестовое диагностирование одномерных однородных структур // Вестник НТУ ХПИ «Автоматика и приборостроение».–Харьков.: НТУ «ХПИ». –2008. –№3.-с.49-57. 16. *Дербунович Л.В., Бережная М.А., Королева Я.Ю., Рыжикова М.Г.* Синтез тестов для однородных структур // Информационно-управляющие системы на железнодорожном транспорте, Харьков.:–2008.–№4.–с.29-33. 17. *Дербунович Л.В., Горлов Ю.В., Татаренко Д.А.* Генераторы тестов на клеточных автоматах для схем встроенного самотестирования // Вісник Національного технічного університету «ХПІ».–Харків: НТУ «ХПІ».–2003.–№21.-с. 59-62. 18. *Дербунович Л.В., Суздаль В.С., Соболев А.В.* Моделирование динамических систем на сетях клеточных автоматов // Весник НТУ ХПІ «Автоматика и приборостроение».–Харьков.–2005.–№7.–52-57 с. 19. *Friedman A.D., Menon P.R.* Fault detection in digital circuits.–New Jersey: Prentice Hall.–1971.–220 p. 20. *Тоценко В.Г.* Алгоритмы технического диагностирования дискретных устройств. –М.: Радио и связь.–1983. –240 с. 21. *Pries W., Thanailakis A., and Card H.C.* Group properties of cellular automata and VLSI applications// IEEE Trans. Comput.–1986.–vol. C-35.–p. 1013–1024. 22. *Hortensius P.D., McLeod R.D., Pries W., Miller D.M. and Card H.C.* Cellular automata–based pseudorandom number generators for built–in self–test. // IEEE Trans. CAD.–1989.–№ 8–p. 842–859.

Поступила в редколлегию 14.11.2008